实验四（3） lsb匹配算法及其性能测试

一 实验目的：LSB匹配算法及其性能测试

二 实验内容：

仿真LSB匹配算法并测试其鲁棒性、透明性

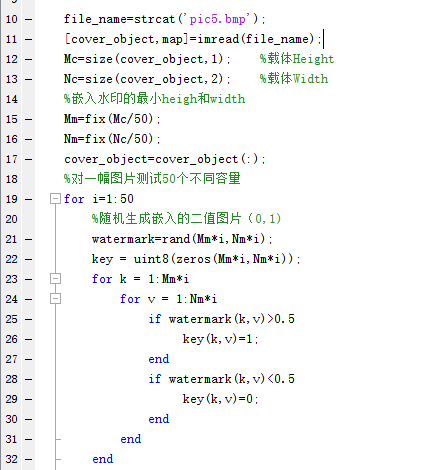
三 实验过程

1. 水印信息嵌入过程

\* 对同一副图像进行操作，得到50种不同容量的含密载体，方便之后绘制psnr曲线

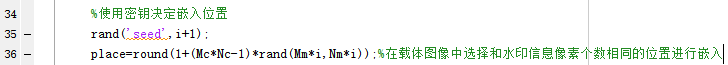
（1）生成不同的水印信息

\* 代码实现：



1. 引入密钥选择在载体中的嵌入位置

\* 代码实现（使用rand()方便生成在指定范围内的数字）：



1. 进行水印信息的嵌入

\* 代码实现：



1. 水印信息提取过程

\* 水印嵌入的逆过程

\* 代码实现：

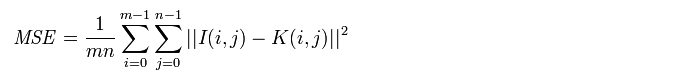


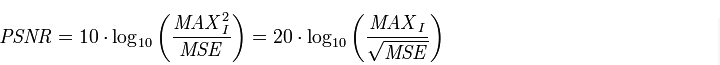
3、LSB算法PSNR~容量测试

（1）目的：测试PSNR与嵌入水印信息量的关系

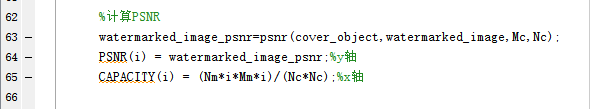
（2）代码实现

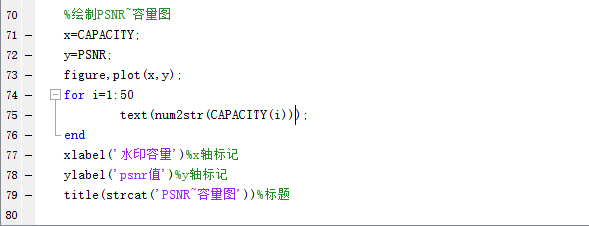
①计算PSNR

\*计算均方差：

\*计算PSNR（MAX1=255）：

②绘制PSNR~容量曲线

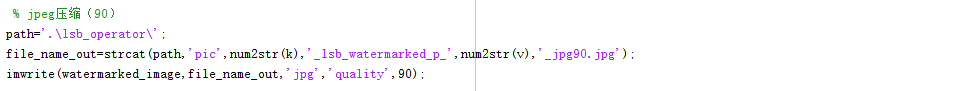




4、LSB替换算法鲁棒性测试

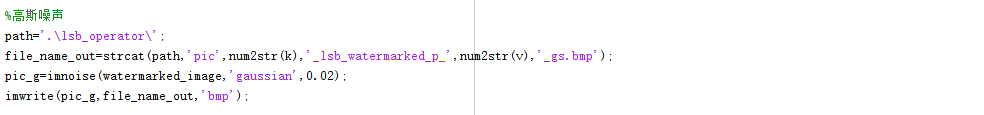
（1）JPEG压缩（q可调）

\*代码实现



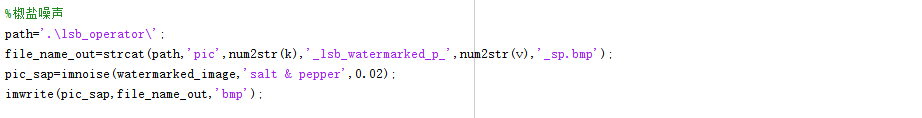
（2）高斯噪声（参数可调）

\*代码实现



（3）椒盐噪声（参数可调）

\*代码实现



1. 实验结果
2. psnr~容量曲线图
3. 图片一

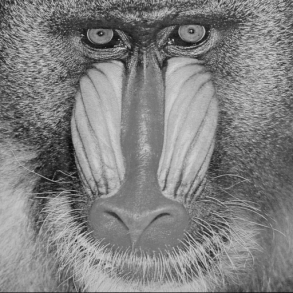


Figure 1. 图片一

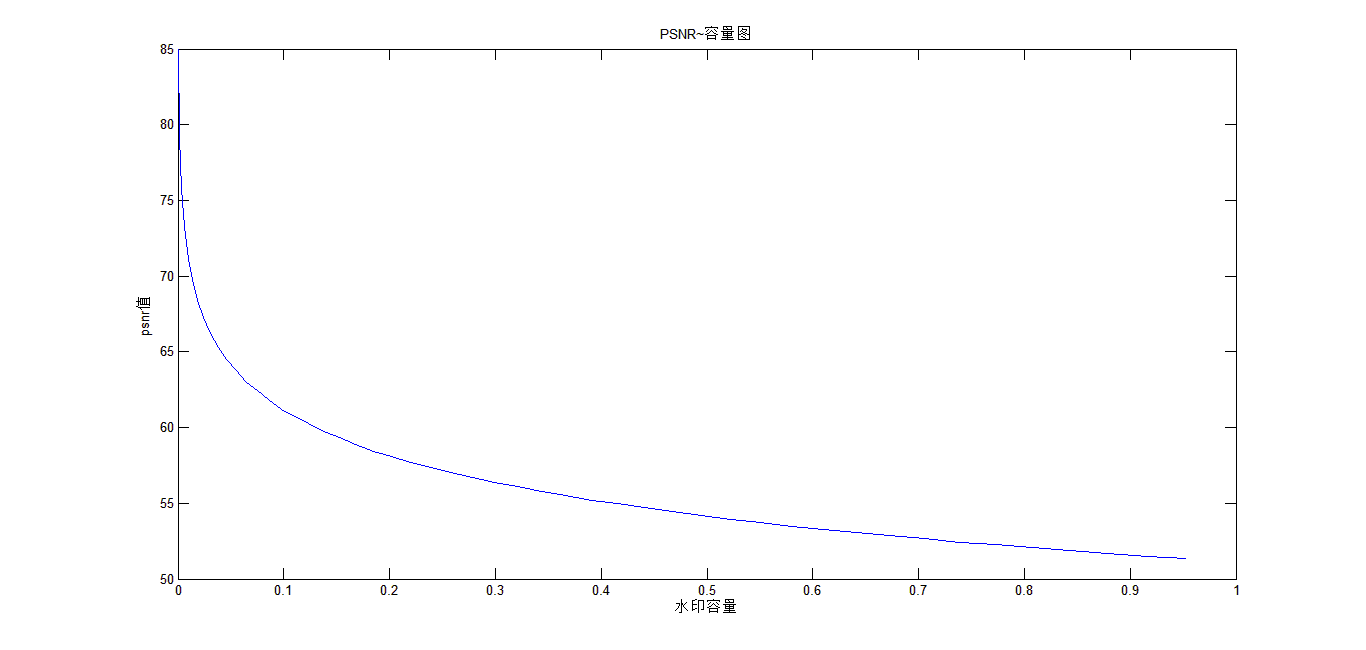


Figure 2. 图片一psnr~容量图

1. 图片二



Figure 3. 图片二

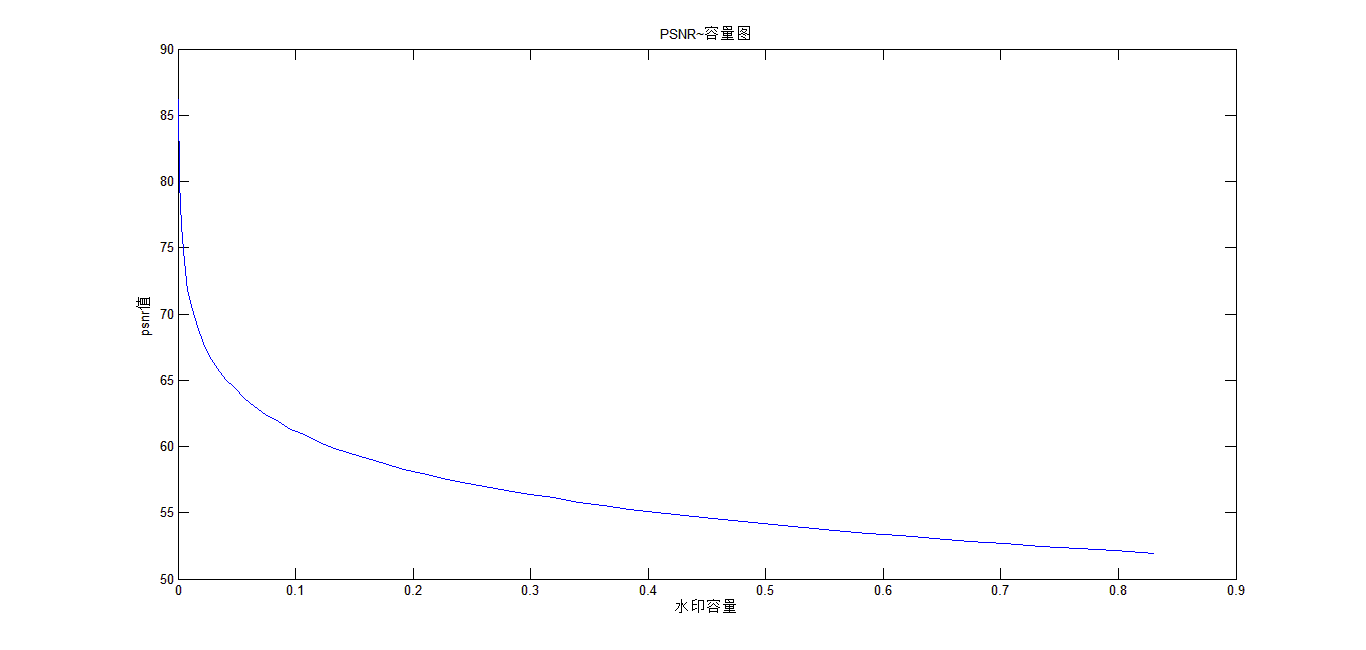


Figure 4. 图片二psnr~容量图

（3）图片三



Figure 5. 图片三

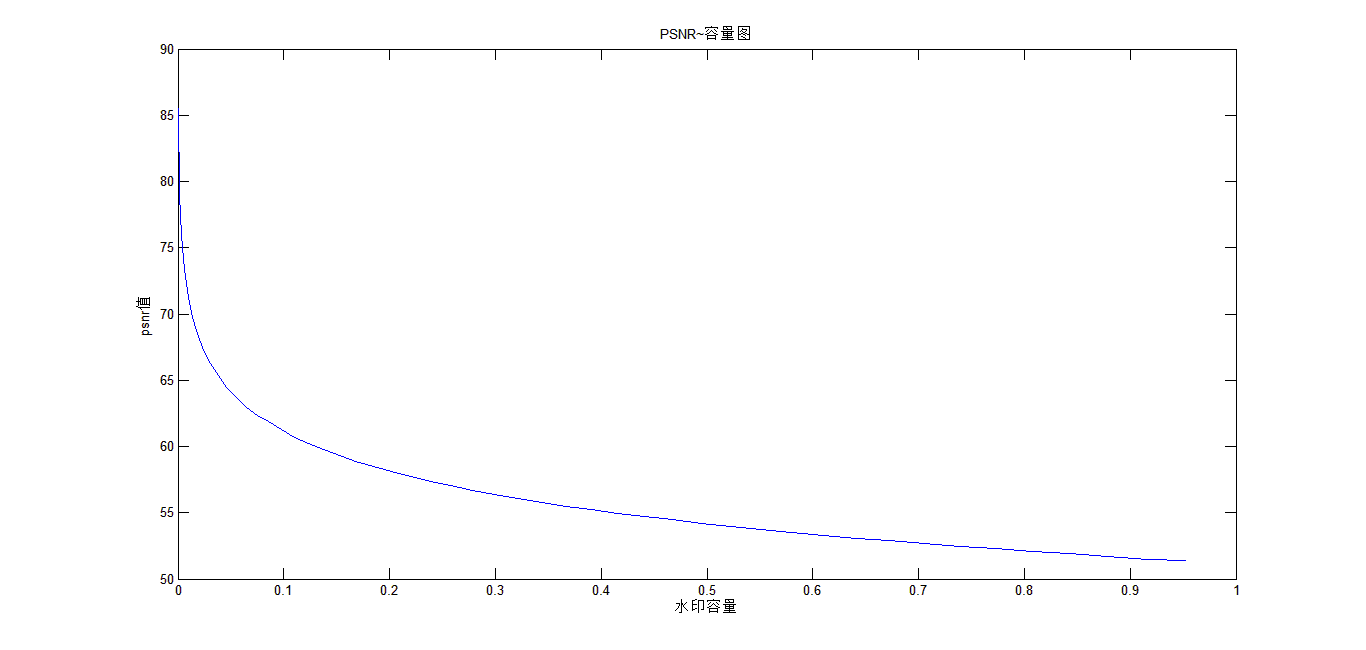


Figure 6. 图片三psnr~容量图

1. 鲁棒性测试（仅展示嵌入容量接近为0.5时测试结果）
2. 图片一

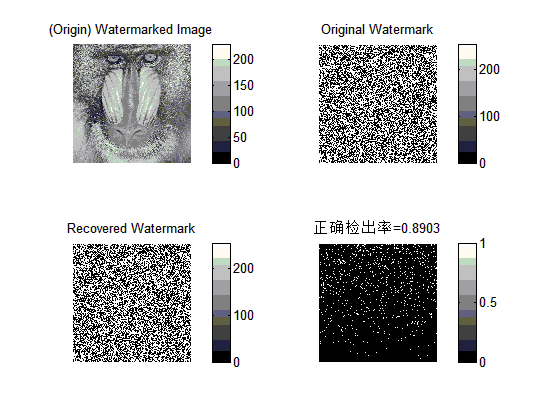


Figure 7. 图片一含密载体未篡改时检出率

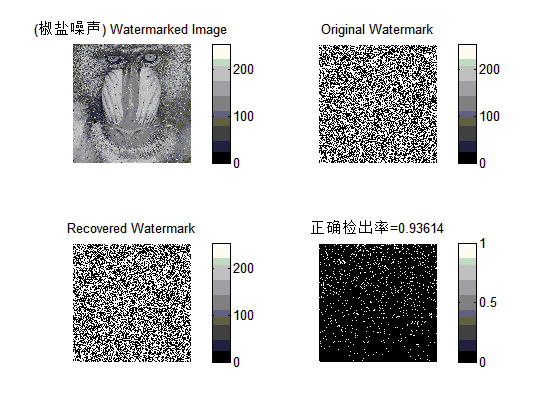


Figure 8. 图片一含密载体jpeg压缩后检出率

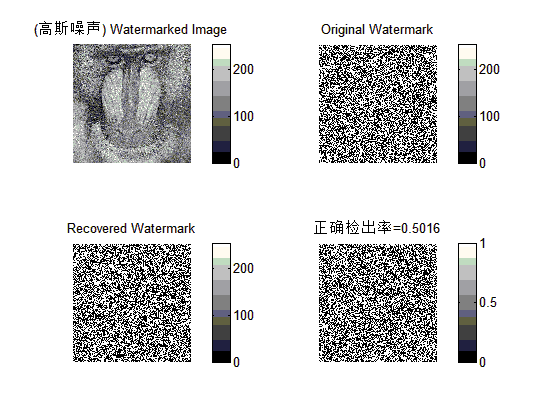


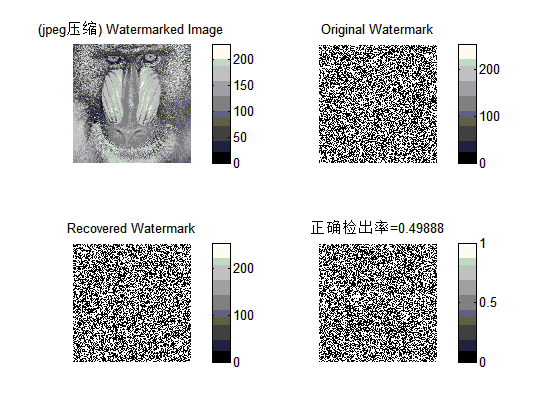
Figure 9. 图片一含密载体加入高斯噪声后检出率

Figure 10. 图片一含密载体加入椒盐噪声后检出率

1. 图片二

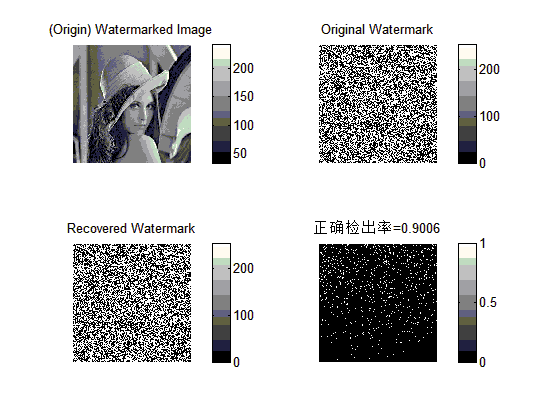


Figure 11. 图片二含密载体未篡改时检出率

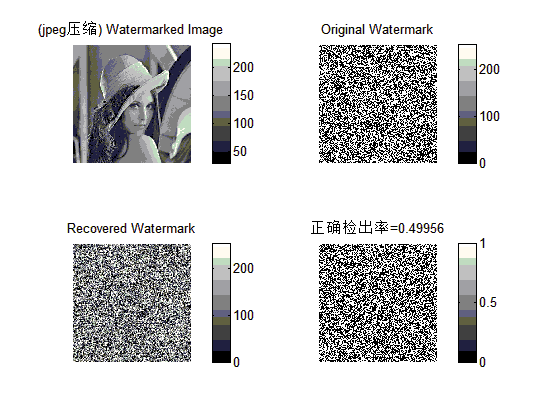


Figure 12. 图片二含密载体jpeg压缩后检出率

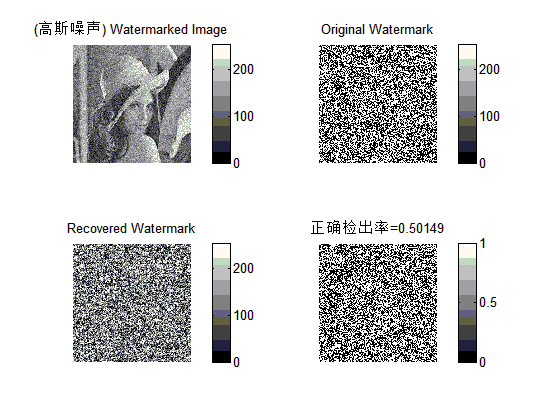


Figure 13. 图片二含密载体加入高斯噪声后检出率

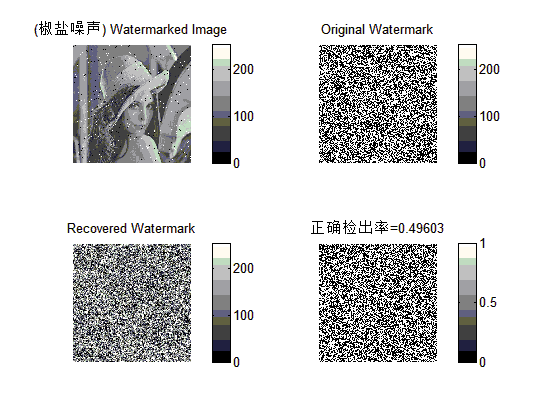


Figure 14. 图片二含密载体加入椒盐噪声后检出率

1. 图片三

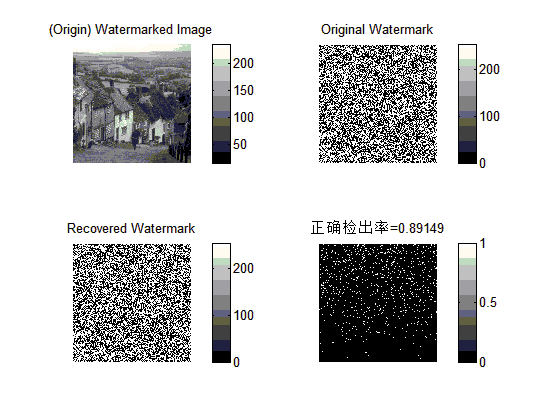


Figure 15. 图片三含密载体未篡改时检出率

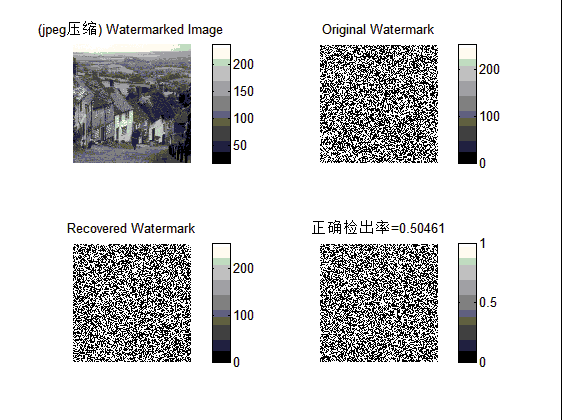


Figure 16. 图片三含密载体jpeg压缩后检出率

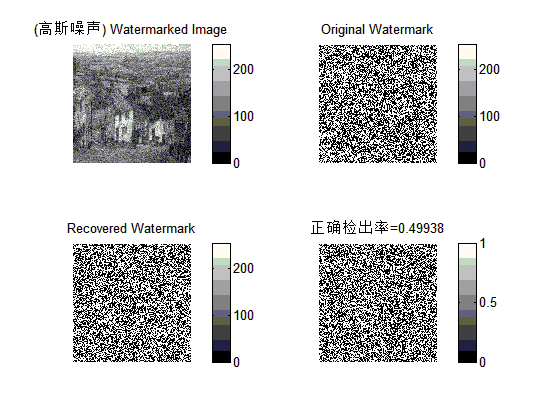


Figure 17. 图片三含密载体加入高斯噪声后检出率

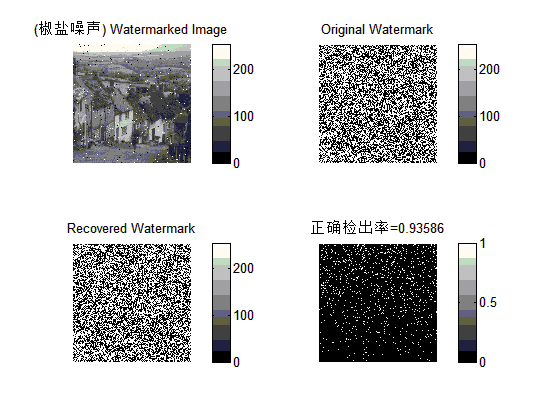


Figure 18. 图片三含密载体加入椒盐噪声后检出率

1. 实验结论

经过实验可以得出，lsb匹配算法简单，易于实现，嵌入信息少时计算速度快，但是嵌入消息较大时，所花时间长。但是对空域的各种操作仍较为敏感，抗JPEG压缩能力弱，鲁棒性差。但通过之前的卡方分析可知，该算法的透明性更好。

1. 附录（实验代码）

代码执行逻辑：

lsb\_match\_embed.m —> pic\_operator.m —> lsb\_match\_extract.m —> robust\_test.m

psnr.m

见附件